NASA Technical Memorandum

NASA TM - 86574

SMALL COMPUTER INTERFACE TO A STEPPER MOTOR

By Fred A. Berry, Jr.

Space Sciences Laboratory
Science and Engineering Directorate

December 1986

(NASA-TM-86574) SMALL COMPUTER INTERFACE TO A STEPPER MOTOR (NASA) 13 p CSCL 09C N87-16970

Unclas
G3/33 43846

National Aeronautics and Space Administration
George C. Marshall Space Flight Center

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
NASA TM- 86574		

4. TITLE AND SUBTITLE

Small Computer Interface to a Stepper Motor

5. REPORT DATE

December 1986

6. PERFORMING ORGANIZATION CODE

ES62

7. AUTHOR(S)

Fred A. Berry, Jr.

8. PERFORMING ORGANIZATION REPORT #

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Space Science Laboratory
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

10. WORK UNIT NO.

11. CONTRACT OR GRANT NO.

12. SPONSORING AGENCY NAME AND ADDRESS

National Aeronautics and Space Administration
Washington, D.C. 20546

13. TYPE OF REPORT & PERIOD COVERED

Technical Memorandum

14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

Prepared by Science and Engineering Directorate, Space Sciences Laboratory

16. ABSTRACT

A Commodore VIC-20 computer has been interfaced with a stepper motor to provide an inexpensive stepper motor controller. Only eight transistors and two integrated circuits compose the interface. The software controls the parallel interface of the computer and provides the four phase drive signals for the motor. Optical sensors control the zeroing of the 12-inch turntable positioned by the controller. The computer calculates the position information and movement of the table and may be programmed in BASIC to execute automatic sequences.

17. KEY WORDS

Stepper Motor, Controller, Computer Control, Interface

18. DISTRIBUTION STATEMENT

Unclassified--Unlimited

Fred A. Berry, Jr.

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	13	NTIS

MSFC - Form 3292 (May 1969)

For sale by National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

		Page
I.	Introduction	1
II.	Description	1
	A. Motor Interface	1
	B. Sensor Interface	1
	C. Computer Program	2
III.	Application	2

LIST OF ILLUSTRATIONS

Figure	Title	Page
1.	Octal buffer interface	3
2.	Stepper motor interface	4
3.	Sensor interface	5

LIST OF TABLES

Table	Title	Page
1.	Program Listing for VIC-20 Computer	6
2.	Program Listing for Commodore 64 and 128 Computers	8

TECHNICAL MEMORANDUM

SMALL COMPUTER INTERFACE TO A STEPPER MOTOR

I. INTRODUCTION

In the past, stepper motors have been controlled by expensive dedicated electronics with little or no programmability or flexibility. A Commodore VIC-20 computer was interfaced with a controller to provide an inexpensive programmable stepper motor controller (see Table 1). A Commodore 64 version of the program is available and may be run on either a C-64 or a C-128 computer operated in C-64 mode (see Table 2). The computer program is written in BASIC and may be adapted to any small computer with a parallel interface. This controller allows the user to both interact with the stepper motor and program in a special sequence of steps for the motor to perform.

The drive circuits of the motor controller, originally designed by Guy N. Brown, serve to control current to the motor and provide level shifting for the computer interface. An interface integrated circuit (IC) provides isolation and line driving capabilities to the circuit. The computer provides both the four phase drive signals for the motor and maintains the position information of the table. The computer may be separated from the motor controller by a distance of twenty or more meters without resorting to special line driver electronics.

The parallel port of the computer is isolated from the line by an identical interface IC. This IC provides the line driving and receiving functions for the VIC-20 parallel port. Two sets of infrared position sensors, coarse and fine, are used by the computer to detect the zero setting of the stepper motor. The coarse sensor detects the zero position of the table while the fine sensor defines the zero position of the drive motor shaft. Two comparator circuits are used to condition the optical sensor outputs for the computer.

II. DESCRIPTION

A. Motor Interface

The computer controls the octal buffer interface and line drivers (see Figure 1). Cables connect the computer interface to the controller containing the other 74LS244 octal buffer IC (see Figure 2). The 2N2222 switching transistors are controlled by the octal buffer. The 2N2222 transistors control the high current motor switching transistors to buffer the motor from the TTL electronics. The field coils of the stepper motor, in turn, are switched directly by 2N6191 high current capacity transistors. Other output transistors may be required to interface to the particular stepping motor used. The 2N6191's may be used safely to switch the phase current for up to 2 amperes.

B. Sensor Interface

The sensors used for this project are infrared emitter LED's and photo-transistors packaged together (see Figure 3). The outputs of the photo-transistors are fed into LM111 comparators. These comparators detect a threshold when a sensor is either blocked or activated. Then the comparator outputs a logic pulse to the buffer which is then sent to the parallel port of the computer for use in the program.

C. Computer Program

The computer program is written in BASIC on a Commodore VIC-20 computer. This computer was chosen for its compact size and low cost. A slight modification to the address for the parallel port will allow the program to function on a Commodore C-64 or C-128 computer. Other modifications would be necessary for use with other brands of computers.

Statements 1-3 (see listing) of the program contain the port assignments with P being the port address value and D being the Data Direction Register. Statement 10 refers to the subprogram in statements 800-890, the operating instruction subroutine. This subprogram contains all the necessary operating instructions and allows the user to use the program without referring to any manual or instruction sheet.

Statements 11-30 form the main program loop. In statements 11-15, the information about direction and distance of rotation of the table is inputted. This is where any custom programming or subroutines might be placed to take advantage of the computer's programmability. Statements 16 and 25-29 take care of steps less than one full revolution of the motor. Statements 22-23 take care of full revolution travel.

Statements 40-90 are the counter-clockwise movement of the stepper motor for less than one full revolution; i.e., they set which bits are on and off for each step. Statements 140-190 do the same thing for the clockwise motion. Statements 331-390 form the subroutine that controls full revolution movement of the motor in the counter-clockwise direction. Statements 391-500 do the same in the clockwise direction.

Each routine determines the sequence of bits to turn the different fields of the stepper motor on and off and the pattern of movement of the motor. Statements 377 and 437 provide for the intervention of the operator to stop a movement in progress. Statements 375 and 475 provide an odometer readout of the number of steps taken. Note that all displays and inputs are in degrees and the number of steps is 1/100 of a degree. This is because the gear ratio of the turntable moved by the stepper motor is 180:1. Other gear ratios and other applications of the controller might use other numbers.

Statements 600-740 are a zeroing subroutine for the turntable. A best direction query in line 600 sets the program to the most efficient direction of returning to zero. This method was chosen over the method where the program keeps up with the best direction because the system might be shut down with the table of zero and get confused on powerup. The subroutine looks for the coarse zero being detected in lines 630 or 710. Lines 632-635 or 715-725 search for the fine zero. Line 640 or 740 returns to the main loop.

III. APPLICATION

The controller is now being used to command a 12-inch turntable. By controlling the stepper motor to within one step, a positional accuracy of ±0.01, has been achieved. This is consistent with the accuracy of the turntable being controlled. Further modifications of the program would allow the user to move the stepper motor in half-steps and increase the accuracy of the positioner. Also, the motor is turned off after each operation to conserve power and allow for cooling. If it is necessary to hold something in position, a holding current could be implemented simply by omitting statements 85, 185, 385, and 450. One may modify the program to include linear and other positional variations. Various sequences might be as easily programmed in so that the motor would carry out a list of instructions. Further enhancements could include an acceleration/deceleration routine, programmed sequences, and motor feedback to the computer.

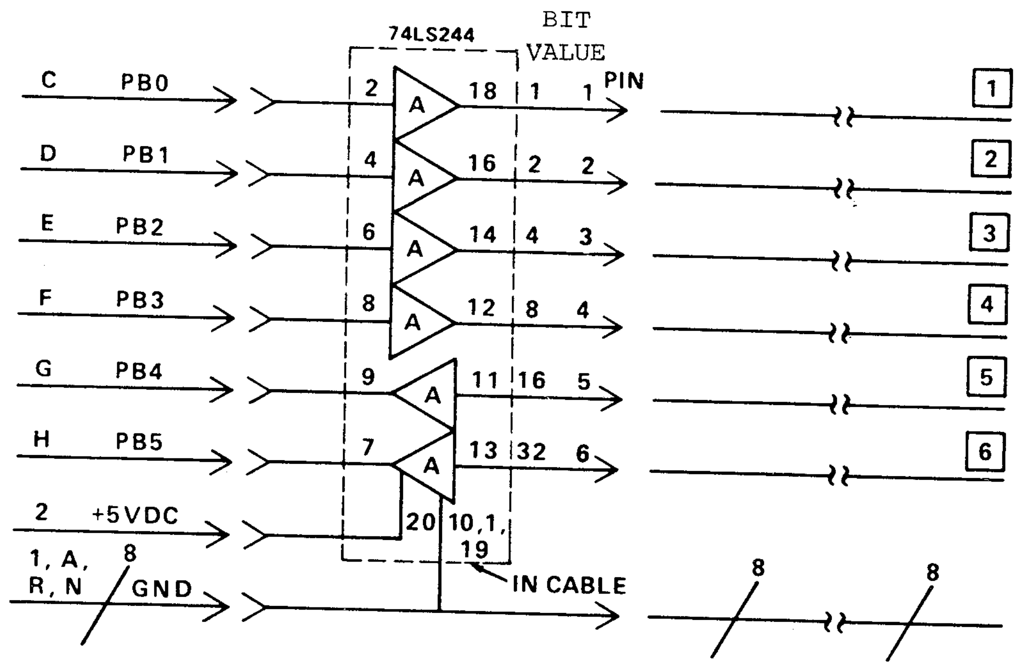
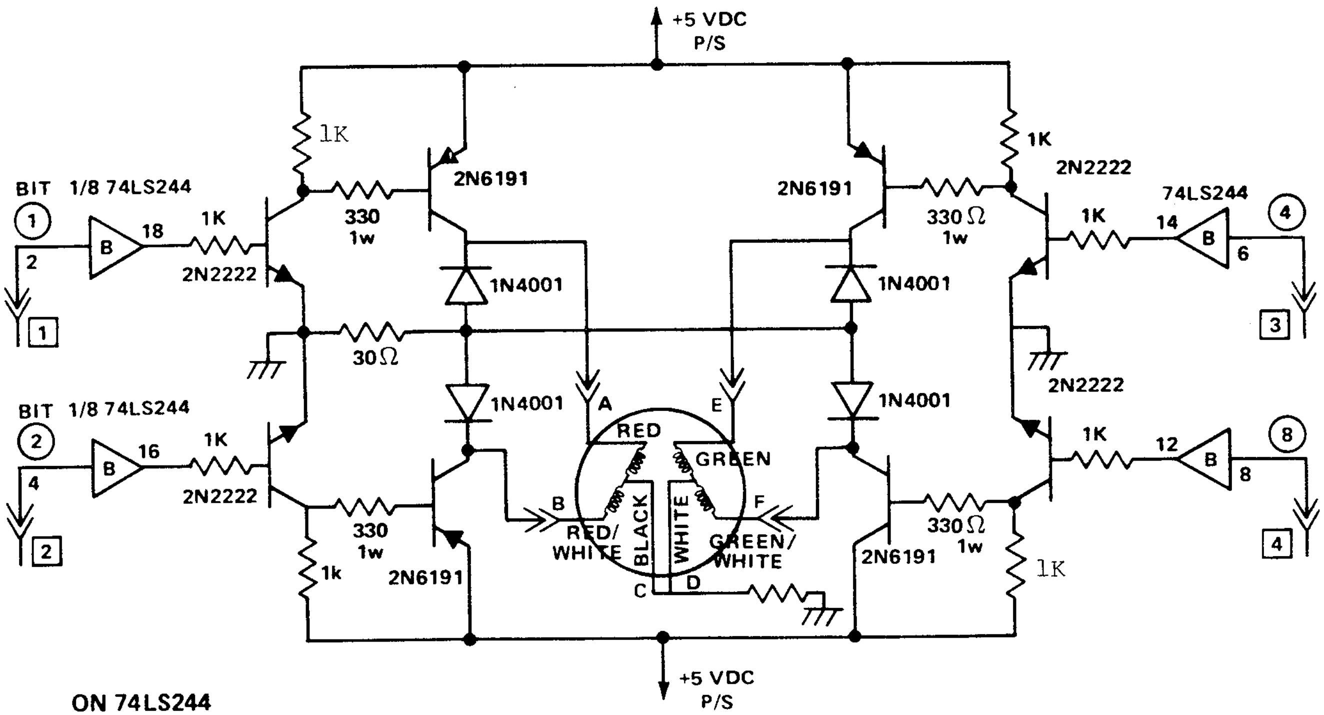
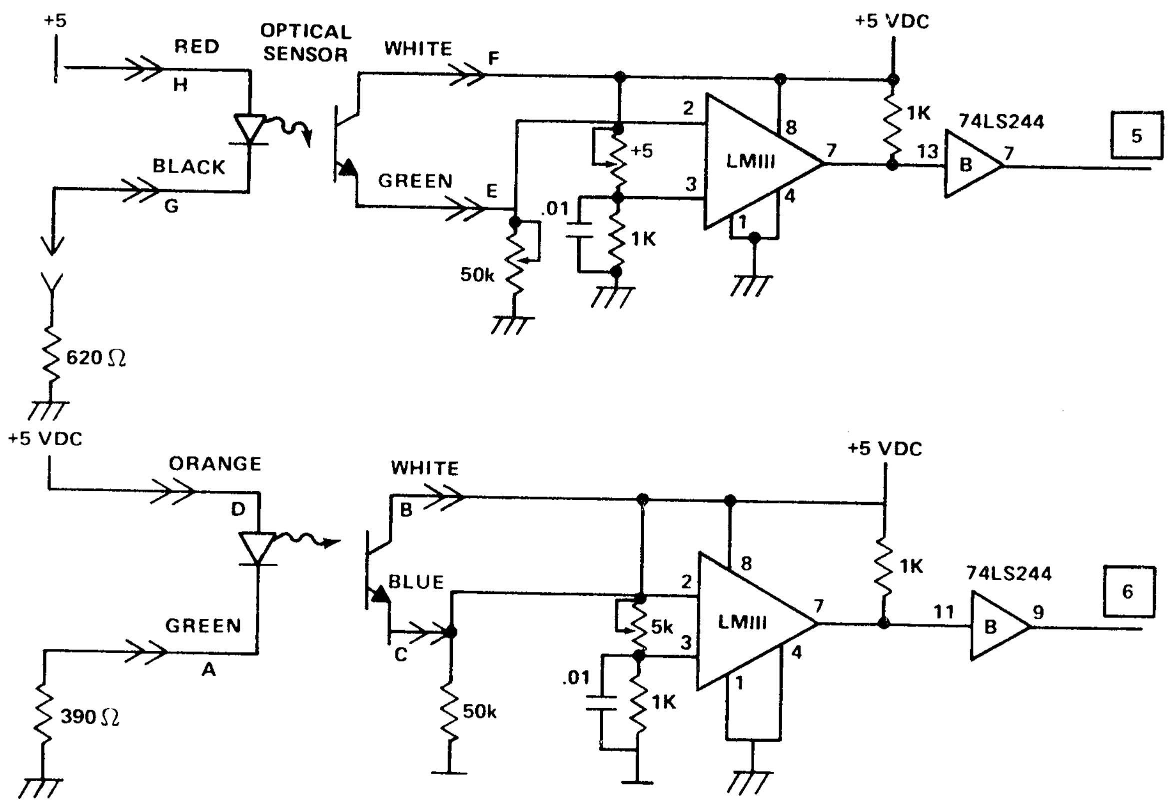


Figure 1. Octal Buffer Interface.



ON 74LS244
PIN
20 +5
10,1,15 GND

Fig. 2. Stepper Motor Interface.



CABLE FOR SENSORS

CONN PIN	CABLE COLOR	SENSOR COLOR
A	WHITE/BLUE	GREEN
B	ORANGE/WHITE	WHITE
C	BLUE/WHITE	BLUE
D	WHITE/ORANGE	ORANGE
E	ORANGE/WHITE	GREEN
F	WHITE/ORANGE	WHITE
G	BLUE/WHITE	BLACK
H	WHITE/BLUE	RED

Fig. 3. Sensor Interface

TABLE 1. PROGRAM LISTING FOR VIC-20 COMPUTER

```
1 P=37136:D=37138:PA=1:Q=15:PO%=0
2 POKED,15
3 POKEP,0
10 GOSUB 800
11 INPUT"WHICH DIR";DI$
12 IFDI$="ZERO"THENPO%=0:GOTO600
13 IFDI$="E"THEN1000
14 PRINT"ROTATION IS IN DECIMAL DEGREES 'XXX.XX'"
15 INPUT"# OF DEGREES";W:T=INT(W*100):Z=T/4-1
16 IFT<4THEN25
22 IFDI$="CW"THENGOSUB391
23 IFDI$="CCW"THENGOSUB331
25 C=(T/4-INT(T/4))*4
26 IFT>4AND C=0THENC=4
28 IFDI$="CCW"THENPO%=PO%-C:ONCGOSUB70,60,50,40:XO=PO%:PRINT
   "TABLE AT";XO/100;"DEG"
29 IFDI$="CW"THENPO%=PO%+C:ONCGOSUB170,160,150,140:XO=PO%:PRINT
   "TABLE AT";XO/100;"DEG"
30 GOTO11
31 REM
40 POKEP,QAND5
50 POKEP,QAND6
60 POKEP,QAND10
70 POKEP,QAND9
85 POKEP,0
90 RETURN
140 POKEP,QAND9
150 POKEP,QAND10
160 POKEP,QAND6
170 POKEP,QAND5
185 POKEP,0
190 RETURN
331 REM
335 FORX=1TOZ
340 POKEP,QAND5
350 POKEP,QAND6
360 POKEP,QAND10
370 POKEP,QAND9
375 PO%=PO%-4:XO=PO%:PRINTXO/100;"DEGREES"
377 IF(PEEK(198))>0THENX=Z
380 NEXT
381 REM
385 POKEP,0
390 PRINT""
437 IF(PEEK(198))>0THENX=Z
440 NEXT
450 POKEP,0
500 PRINT"{CLR}END":RETURN
600 INPUT"BEST DIR";DI$
610 IFDI$="CW"THEN700
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 1 (concluded)

```
620 DI$="CCW"
625 REMPRINT(PEEK(P)AND32)
630 IF(PEEK(P)AND32)<>0THENGOSUB40:GOTO625
632 IF(PEEK(P)AND16)=0THENGOSUB40
635 IF(PEEK(P)AND16)=0THEN632
640 GOTO11
700 REMPRINT(PEEK(P)AND32)
710 IF(PEEK(P)AND32)<>0THENGOSUB140:GOTO700
715 IF(PEEK(P)AND16)=0THENGOSUB140
725 IF(PEEK(P)AND16)=0THEN715
740 GOTO11
800 PRINT"{CLR}INSTRUCTIONS FOR USE OF VIC 20 STEPPER MOTOR
    INTERFACE"
810 PRINT"INPUT 'CW' TO MOVE THE TABLE CLOCKWISE"
820 PRINT"INPUT 'CCW' TO MOVE THE TABLE COUNTERCLOCKWISE"
830 PRINT"INPUT 'ZERO' TO MOVE THE TABLE TO IT'S INITIAL
    POSITION"
835 PRINT"A PREFERRED DIRECTION SHOULD BE INPUTTED TO AID THE
    ZEROING ROUTINE"
840 PRINT"INPUT 'E' TO END THE PROGRAM
850 PRINT
890 RETURN
1000 END
1010 OPEN15,8,15,"S0:OS"
1015 CLOSE15
1020 SAVE"OS",8
1030 OPEN15,8,15,"S0:ODOSTEPPER"
1035 SAVE"ODOSTEPPER",8
1060 CLOSE15
1080 END
```

TABLE 2. PROGRAM LISTING FOR COMMODORE 64 AND 128 COMPUTERS

```
1 P=56577:D=56579:PA=1:Q=15:PO%=0
2 POKED,15
3 POKEP,0
10 GOSUB 800
11 INPUT"WHICH DIR";DI$
12 IFDI$="ZERO"THENPO%=0:GOTO600
13 IFDI$="E"THEN1000
14 PRINT"ROTATION IS IN DECIMAL DEGREES 'XXX.XX'"
15 INPUT"# OF DEGREES";W:T=INT(W*100):Z=T/4-1
16 IFT<4THEN25
22 IFDI$="CW"THENGOSUB391
23 IFDI$="CCW"THENGOSUB331
25 C=(T/4-INT(T/4))*4
26 IFT>4AND C=0THENC=4
28 IFDI$="CCW"THENPO%=PO%-C:ONCGOSUB70,60,50,40:XO=PO%:PRINT
   "TABLE AT";XO/100;"DEG"
29 IFDI$="CW"THENPO%=PO%+C:ONCGOSUB170,160,150,140:XO=PO%:PRINT
   "TABLE AT";XO/100;"DEG"
30 GOTO11
31 REM
40 POKEP,QAND5
50 POKEP,QAND6
60 POKEP,QAND10
70 POKEP,QAND9
85 POKEP,0
90 RETURN
140 POKEP,QAND9
150 POKEP,QAND10
160 POKEP,QAND6
170 POKEP,QAND5
185 POKEP,0
190 RETURN
331 REM
335 FORX=1TOZ
340 POKEP,QAND5
350 POKEP,QAND6
360 POKEP,QAND10
370 POKEP,QAND9
375 PO%=PO%-4:XO=PO%:PRINTXO/100;"DEGREES"
377 IF(PEEK(198))>0THENX=Z
380 NEXT
381 REM
385 POKEP,0
390 PRINT""
437 IF(PEEK(198))>0THENX=Z
440 NEXT
450 POKEP,0
500 PRINT"{CLR}END":RETURN
600 INPUT"BEST DIR";DI$
610 IFDI$="CW"THEN700
620 DI$="CCW"
```

TABLE 2 (concluded)

```
625 REMPRINT(PEEK(P)AND32)
630 IF(PEEK(P)AND32)<>0THENGOSUB40:GOTO625
632 IF(PEEK(P)AND16)=0THENGOSUB40
635 IF(PEEK(P)AND16)=0THEN632
640 GOTO11
700 REMPRINT(PEEK(P)AND32)
710 IF(PEEK(P)AND32)<>0THENGOSUB140:GOTO700
715 IF(PEEK(P)AND16)=0THENGOSUB140
725 IF(PEEK(P)AND16)=0THEN715
740 GOTO11
800 PRINT"{CLR}INSTRUCTIONS FOR USE OF VIC 20 STEPPER MOTOR
    INTERFACE"
810 PRINT"INPUT 'CW' TO MOVE THE TABLE CLOCKWISE"
820 PRINT"INPUT 'CCW' TO MOVE THE TABLE COUNTERCLOCKWISE"
830 PRINT"INPUT 'ZERO' TO MOVE THE TABLE TO IT'S INITIAL
    POSITION"
835 PRINT"A PREFERRED DIRECTION SHOULD BE INPUTTED TO AID THE
    ZEROING ROUTINE"
840 PRINT"INPUT 'E' TO END THE PROGRAM
850 PRINT
890 RETURN
1000 END
1010 OPEN15,8,15,"S0:OS"
1015 CLOSE15
1020 SAVE"OS",8
1030 OPEN15,8,15,"S0:ODOSTEPPER"
1035 SAVE"ODOSTEPPER",8
1060 CLOSE15
1080 END
```

APPROVAL

SMALL COMPUTER INTERFACE TO A STEPPER MOTOR

By Fred A. Berry, Jr.

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

E. A. Tandberg-Hanssen

E. A. TANDBERG-HANSSEN
Deputy Director, Space Science Laboratory

☆U.S. GOVERNMENT PRINTING OFFICE 1987 730-067/40051